

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-353140

(43)Date of publication of application : 19.12.2000

(51)Int.Cl.

G06F 13/00  
G06F 13/14

(21)Application number : 11-165567

(71)Applicant : CANON INC

(22)Date of filing : 11.06.1999

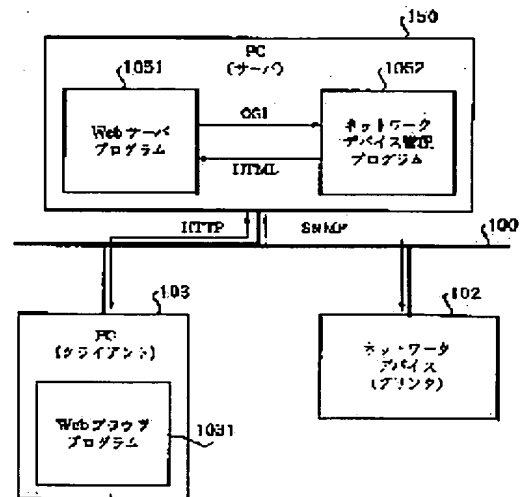
(72)Inventor : KAWASHIMA MASANORI

## (54) DEVICE AND METHOD FOR MANAGING NETWORK

### (57)Abstract:

**PROBLEM TO BE SOLVED:** To obtain network management device and method capable of easily changing an HTML document, its template or the like by preventing the increment of traffic between devices and the increment of load in each device and eliminating the necessity of managing the constitution of a set format object existing in a set page.

**SOLUTION:** At the time of receiving a CGI request from a browser in order to change the set value (MIB information) of a network device 102 by a web browser 1031, a network management program 1052 reads out the MIB information from the target device 102 and prepares an HTML document in accordance with a prescribed template. In this case, settable MIB information is integrated into the document as two data, i.e., hidden type data and settable type data. After the end of setting, the browser 1031 transmits the hidden type data and the set data to a network management program 1052. The program 1052 compares the set data with the transmitted data and sets up the data of a different value in the device 102.



## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-353140

(P2000-353140A)

(43) 公開日 平成12年12月19日 (2000.12.19)

(51) Int.Cl. <sup>7</sup>	識別記号	F I	テーマコード*(参考)
G 0 6 F 13/00	3 5 7	G 0 6 F 13/00	3 5 7 A 5 B 0 1 4
	3 5 3		3 5 3 B 5 B 0 8 9
13/14	3 3 0	13/14	3 3 0 A

審査請求 未請求 請求項の数11 O L (全 31 頁)

(21) 出願番号 特願平11-165587

(22) 出願日 平成11年6月11日 (1999.6.11)

(71) 出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(72) 発明者 川島 正徳

東京都大田区下丸子3丁目30番2号 キヤ  
ノン株式会社内

(74) 代理人 100076428

弁理士 大塚 康德 (外2名)

Fターム(参考) 5B014 EB03 HC02 HC11

5B089 GA13 GA21 GB02 HA06 JA35

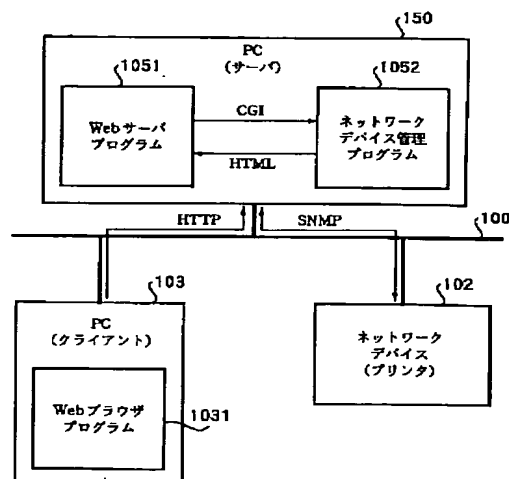
JB02 KA07 KB10

(54) 【発明の名称】 ネットワーク管理装置および方法

(57) 【要約】

【課題】 デバイス情報の変更時、変更されていない設定値も設定値としてネットワークデバイスに送信してしまう。

【解決手段】 Webブラウザ1031によりネットワークデバイスの設定値(MIB情報)を変更するために、ネットワーク管理プログラム1052は、ブラウザからCGIリクエストを受けると、対象デバイス102からMIB情報を読み出し、所定のテンプレートに従ってHTML文書を作成する。その際、設定可能なMIB情報は、隠し形式のデータと設定可能な形式のデータの2つのデータとして文書に組み込まれる。ブラウザは、設定終了後、隠し形式のデータと設定されたデータをネットワーク管理プログラム1052に送信する。ネットワーク管理プログラム1052は、設定データと送信したデータとを比較し、違う値のデータをデバイス102に設定する。



1

## 【特許請求の範囲】

【請求項1】 ネットワークに接続されたネットワークデバイスを管理するネットワーク管理方法であって、デバイスからそのデバイスの設定情報を取得する情報取得工程と、  
前記設定情報を設定し直すための文書データを、前記設定情報を書き換え不可能な形式で含むように作成する作成工程と前記作成工程において作成された文書データに基づいて設定し直された設定情報を含むデータを受信し、前記書き換え不可能な形式で含ませたデータと比較 10 する比較工程と、

前記比較工程による比較の結果、設定し直された設定情報に限って前記デバイスに対して設定する設定工程とを備えることを特徴とするネットワーク管理方法。

【請求項2】 前記文書データはHTML形式の文書データであり、前記ネットワークに接続された端末において表示及び設定し直しが行われることを特徴とする請求項1に記載のネットワーク管理方法。

【請求項3】 サーバから読み込んだ情報をキャッシュするキャッシュ機能を有する端末において、サーバから 20 の情報を変更して再表示させるネットワーク管理方法であって、  
前記端末に送信する情報に含まれる、該情報アクセスするためのリンク情報に、送信する毎に固有な情報を付加する工程と、

前記固有な情報が付加されたリンク情報を含む情報を、前記端末に送信する工程とを備えることを特徴とするネットワーク管理方法。

【請求項4】 前記付加工程は、HTML文書データに含まれるリンク情報として、送信する毎に固有な情報を 30 付加することを特徴とする請求項3に記載のネットワーク管理方法。

【請求項5】 前記端末に送信する情報は、前記端末により設定される、ネットワークデバイスの設定情報であることを特徴とする請求項3に記載のネットワーク管理方法。

【請求項6】 前記送信する毎に固有な情報は、時刻を示す情報であることを特徴とする請求項3に記載のネットワーク管理方法。

【請求項7】 前記送信する毎に固有な情報は、所定数 40 を法として漸増する数であることを特徴とする請求項3に記載のネットワーク管理方法。

【請求項8】 ネットワークに接続されたネットワークデバイスを管理するネットワーク管理装置であって、デバイスからそのデバイスの設定情報を取得する情報取得手段と、  
前記設定情報を設定し直すための文書データを、前記設定情報を書き換え不可能な形式で含むように作成する作成手段と前記作成手段において作成された文書データに基づいて設定し直された設定情報を含むデータを受信 50

2

し、前記書き換え不可能な形式で含ませたデータと比較する比較手段と、

前記比較手段による比較の結果、設定し直された設定情報に限って前記デバイスに対して設定する設定手段とを備えることを特徴とするネットワーク管理装置。

【請求項9】 サーバから読み込んだ情報をキャッシュするキャッシュ機能を有する端末において、サーバからの情報を変更して再表示させるネットワーク管理装置であって、

前記端末に送信する情報に含まれる、該情報アクセスするためのリンク情報に、送信する毎に固有な情報を付加する手段と、

前記固有な情報が付加されたリンク情報を含む情報を、前記端末に送信する手段とを備え、前記端末は前記リンク情報をたどって変更後の情報を獲得することを特徴とするネットワーク管理装置。

【請求項10】 コンピュータにより、デバイスからそのデバイスの設定情報を取得する情報取得手段と、

前記設定情報を設定し直すための文書データを、前記設定情報を書き換え不可能な形式で含むように作成する作成手段と前記作成手段において作成された文書データに基づいて設定し直された設定情報を含むデータを受信し、前記書き換え不可能な形式で含ませたデータと比較する比較手段と、  
前記比較手段による比較の結果、設定し直された設定情報に限って前記デバイスに対して設定する設定手段とを実現するためのコンピュータプログラムを格納するコンピュータ可読の記憶媒体。

【請求項11】 コンピュータにより、サーバから読み込んだ情報をキャッシュするキャッシュ機能を有する端末に対して送信する情報に含まれる、該情報アクセスするためのリンク情報に、送信する毎に固有な情報を付加する手段と、

前記固有な情報が付加されたリンク情報を含む情報を、前記端末に送信する手段とを実現するためのコンピュータプログラムを格納するコンピュータ可読の記憶媒体。

## 【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、例えばコンピュータネットワークに接続されたデバイス等を管理するためのネットワーク管理装置および方法に関する。

【0002】

【従来の技術】コンピュータは、ローカルエリアネットワーク (Local Area Network、LAN) によって相互に接続することができる。LANは、ビルフロアまたはビル全体、ビル群 (構内)、地域、あるいはさらに大きいエリアにわたる、最大のものでは世界的システムにおよぶ更に大きなシステムに組み込むよう相互に接続することができる。各々のLANは、多様なハードウェア相

互接続技術といくつものネットワークプロトコルをもつことがある。

【0003】他と切り離された簡単なLANは、個々のユーザが管理することができる。すなわち、ユーザが機器を取り替えたり、ソフトウェアをインストールしたり、問題点を診断することができる。

【0004】一方、規模の大きい複雑なLANや相互接続された大きなLANグループは「管理」を必要とする。「管理」とは、人間のネットワークデバイス管理者とその管理者が使用するソフトウェアの両方による管理<sup>10</sup>を意味する。本明細書においては、「管理」とはシステム全体を管理するためのソフトウェアによる管理を意味し、「ユーザ」とはネットワークデバイス管理ソフトウェアを使用する人を意味するものとする。このユーザは、通常、システム管理責任者である。ユーザは、ネットワークデバイス管理ソフトウェアを使うことによって、ネットワーク上で管理データを得て、このデータを変更することができる。

【0005】大規模ネットワークシステムは、通常、機器の増設と除去、ソフトウェアの更新、および問題の検<sup>20</sup>出などを絶えず行うことが必要な動的システムである。一般に、様々な人が所有する、様々な業者から供給される様々なシステムがあるだろう。

【0006】以下、管理が必要な大規模なネットワークの一例について説明する。

【0007】図1は、プリンタをネットワークに接続するためのネットワークボード(NB)101を、開放型アーキテクチャをもつプリンタ102につなげた場合を示す図である。NB101は、ローカルエリアネットワーク(LAN)100に、同軸コネクタをもつEthernet<sup>30</sup>インタフェース10Base-2や、RJ-45をもつ10Base-TなどのLANインタフェースを介して接続される。

【0008】PC103、PC104などの、複数のパーソナルコンピュータ(PC)もまた、LAN100に接続されており、ネットワークオペレーティングシステムの制御のもと、これらのPCはNB101と通信することができる。従って、PCの1つ、例えばPC103を、ネットワークデバイス管理用に使用することができる。また、PCに、PC104に接続されているプリンタ105のようなローカルプリンタを接続してもよい。<sup>40</sup>

【0009】LAN100には、ファイルサーバ106が接続されており、ファイルサーバ106は大容量(例えば、100億バイト)のネットワークディスク107に記憶されたファイルへのアクセスを管理する。

【0010】プリントサーバ108は、接続された複数のプリンタ109、または遠隔地にあるプリンタ105などのプリンタに印刷を行わせる。また、他の図示しない周辺機器をLAN100に接続してもよい。

【0011】また、LAN100には、Webサーバ150が接続されており、Webサーバ150上にインス<sup>50</sup>

トールされたネットワークデバイス管理ソフトウェアが生成したHTML(ハイパーテキストマークアップランゲージ)ドキュメントを、PC103上にインストールされたWebブラウザを使って表示したり、PC103上のWebブラウザ上で行ったプリンタの設定を、Webサーバ150上のネットワークデバイス管理ソフトウェアを介して特定のプリンタに送信することもできる。なお、HTML文書は、文書のレイアウトやファイルの定義、外部アプリケーションの呼出しやスクリプト言語による記述などを、タグと呼ばれる所定の文字列により表した文書データである。

【0012】更に詳しくは、図1に示すネットワークは、様々なネットワークメンバー間で効率良く通信を行うために、NovellやUNIXのソフトウェアなどのネットワークソフトウェアを使用することができる。どのネットワークソフトウェアを使用することも可能であるが、例えば、Novell社のNetWare(Novell社の登録商標)ソフトウェアを使用することができる。このソフトウェアパッケージに関する詳細な説明は、NetWareパッケージに同梱されているオンラインドキュメンテーションに記載されている。これは、Novell社からNetWareパッケージと共に購入することができる。

【0013】簡潔に説明すると、ファイルサーバ106は、LANメンバー間でファイルの受信や、記憶、キューイング、キャッシング、および送信を行う、ファイル管理部としての役割を果たす。例えば、PC103およびPC104のそれぞれにより作成されたデータファイルは、ファイルサーバ106へ送られ、ファイルサーバ106は、これらのデータファイルを順に並べ、そしてプリントサーバ108からのコマンドに従って、並べられたデータファイルをプリンタ109の1つに送信する。

【0014】PC103とPC104は、それぞれデータファイルの生成や、生成したデータファイルのLAN100への送信や、また、LAN100からのファイルの受信や、更に、それらのファイルの表示および/または処理を行うことができる、一般的なPCにより構成される。図1にパーソナルコンピュータ機器が示されているが、ネットワークソフトウェアを実行するのに適切であるような、他のコンピュータ機器を含んでもよい。例えば、UNIXのソフトウェアを使用している場合に、UNIXワークステーションをネットワークに含んでも良く、これらのワークステーションは、適切な状況下で、図示されているPCと共に使用することができる。

【0015】通常、LAN100などのLANは、1つの建物内の1つの階または連続した複数の階でのユーザグループ等の、比較的ローカルなユーザグループにサービスを提供する。例えば、ユーザが他の建物や他県に居るなど、あるユーザが他のユーザから離れるに従って、ワイドエリアネットワーク(Wide Area Network、WAN)を作ってもよい。WANは、基本的には、いくつか

5

のLANをサービス総合デジタルネットワーク (Integrated Services Digital Network、ISDN) などの高速デジタル線で接続することにより形成された集合体である。従って、図1に示すように、LAN100、LAN110、LAN120とは、モデム/トランスポンダ130およびバックボーン140を介して接続され、WANを形成する。

【0016】それぞれのLANは、専用のPCを含み、また、必要に応じて、ファイルサーバやプリントサーバを含むこともある。図1に示すように、LAN110<sup>10</sup>は、PC111、PC112、ファイルサーバ113、ネットワークディスク114、プリントサーバ115、プリンタ116を含む。対照的に、LAN120は、PC121とPC122のみを含む。LAN100、LAN110、およびLAN120に接続されている機器は、WAN接続を介して他のLANに接続された機器の機能にアクセスすることができる。

【0017】このような大規模ネットワークシステムを構成するネットワーク上のデバイスを管理するための方法として、これまでにいくつかの試みが数多くの標準機<sup>20</sup>関でなされている。国際標準化機構 (ISO) は開放型システム間相互接続 (Open System Interconnection、OSI) モデルと呼ばれる汎用基準フレームワークを提供した。ネットワークデバイス管理プロトコルのOSIモデルは、共通管理情報プロトコル (Common Management Information Protocol、CMIP) と呼ばれる。CMIPはヨーロッパの共通ネットワークデバイス管理プロトコルである。

【0018】また近年では、より共通性の高いネットワークデバイス管理プロトコルとして、簡易ネットワーク<sup>30</sup>デバイス管理プロトコル (Simple Network Management protocol、SNMP) と呼ばれるCMIPに関連する一変種のプロトコルがある。 (「TCP/IPネットワークデバイス管理入門実用的な管理をめざして」M. T. ローズ=著/西田竹志=訳 (株) トッパン発行1992年8月20日初版を参照)。

【0019】このSNMPネットワークデバイス管理技術によれば、ネットワークデバイス管理システムには、少なくとも1つのネットワークデバイス管理ステーション (Network-Device Management Station、NMS)、<sup>40</sup> 各々がエージェントを含むいくつかの管理対象ノード、および管理ステーションやエージェントが管理情報を交換するために使用するネットワークデバイス管理プロトコルが含まれる。ユーザは、NMS上でネットワークデバイス管理ソフトウェアを用いて管理対象ノード上のエージェントソフトウェアと通信することにより、ネットワーク上のデータを得、またデータを変更することができる。

【0020】ここでエージェントとは、各々のターゲット装置についてのバックグラウンドプロセスとして走る<sup>50</sup>

6

ソフトウェアである。ユーザがネットワーク上の装置に対して管理データを要求すると、管理ソフトウェアはオブジェクト識別情報を管理バケットまたはフレームに入れてターゲットエージェントへ送り出す。エージェントは、そのオブジェクト識別情報を解釈して、そのオブジェクト識別情報に対応するデータを取り出し、そのデータをバケットに入れてユーザに送り返す。時には、データを取り出すために対応するプロセスが呼び出される場合もある。

【0021】またエージェントは、自分の状態に関する管理データをデータベースの形式で保持している。このデータベースのことを、管理情報ベース (Management Information Base、MIB) と呼ぶ。MIBは木構造のデータ構造をしており、全てのノードが一意に番号付けされている。このノードの識別子のことを、オブジェクト識別子 (OBJECT IDENTIFIER) と呼ぶ。

【0022】このMIBの構造は、管理情報構造 (Structure of Management Information、SMI) と呼ばれ、RFC1155 Structure and Identification of Management Information for TCP/IP-based Internets で規定されている。

【0023】尚、本明細書において、ネットワークデバイスの管理データと、MIBのオブジェクト識別子に割り当てられた情報、MIB情報、は同等のものを指す。

【0024】次に、SNMPプロトコルについて簡単に説明する。ネットワークデバイス管理ユーティリティソフトウェアが動作しているPC (以下、マネージャという) とSNMPエージェントが動作している管理対象ネットワークデバイス (以下、エージェントという) とは、SNMPプロトコルを用いて通信を行う。SNMPプロトコルには5種類のコマンドがあり、それぞれGet-request、Get-next-request、Get-response、Set-request、Trapと呼ばれる。

【0025】Get-requestおよびGet-next-requestは、マネージャがエージェントのMIBオブジェクトの値 (MIB情報) を取得するために、マネージャがエージェントに対して送出するコマンドである。このコマンドを受け取ったエージェントは、MIBオブジェクトの値をマネージャに通知するために、マネージャに対してGet-responseコマンドを送出する。

【0026】Set-requestは、マネージャがエージェントのMIBオブジェクトの値を設定するために、マネージャがエージェントに対して送出するコマンドである。

【0027】このコマンドを受け取ったエージェントは、設定結果をマネージャに通知するために、マネージャに対してGet-responseコマンドを送出する。

【0028】Trapは、エージェントが自分自身の状態の変化をマネージャに対して通知するために、エージェントがマネージャに対して送出するコマンドである。

【0029】SNMPエージェントはPCやプリンタ102に接続されているネットワークボード(NB)101上で動作し、SNMPマネージャとなるネットワークデバイス管理ソフトウェアはPCで動作するシステムがよく知られている。しかし近年のインターネットの普及により、専用のネットワークデバイス管理ソフトウェアをクライアントPC毎に動作させるのではなく、ネットワークデバイス管理ソフトウェアはサーバ上で動作し、さらにユーザインタフェースとしてWWW(World Wide Web)を用いたシステムもでている。

【0030】次に図2を用いて一般的なWWWシステムの動作およびWWWシステムに基づいたSNMP管理プログラムの動作の概要を説明する。

【0031】PC150ではWebサーバプログラム1501が動作しており、PC150のディスクにはHTMLを用いて記述されている多数のページデータ(HTMLドキュメントそのもの、またはHTMLドキュメントを生成するためのテンプレート)が格納されている。

【0032】PC103上で動作するWebブラウザプログラム1031は、ユーザにより指定されたページを表示するために、PC150上で動作するWebサーバプログラム1051に対してCGI(Common Gateway Interface)の規定に従ったリクエスト(CGIリクエスト)を発行することにより、指定されたページの取得を要求する。

【0033】Webサーバプログラム1051は、Webブラウザプログラム1031からの要求への応答として、指定されたページデータ(HTMLドキュメント)を返す。Webブラウザプログラム1031は取得したページデータを解析してその記述に従いページを表示する。

【0034】Webサーバプログラム1051は、Webブラウザプログラム1031からのページ取得要求中にCGIを経由した要求が含まれる場合、CGI所定の方法により外部スクリプトやプログラムを起動し、Webブラウザプログラム1031からの要求への応答のためのページデータを受け取り、それをWebブラウザプログラム1031へ返す。

【0035】次に、上記CGIで起動される外部プログラムが、本願のようなネットワークデバイス管理プログラムの場合を説明する。

【0036】Webサーバプログラム1051によりCGIで起動されたネットワークデバイス管理プログラム1052は、SNMPを用いてネットワークに接続されているデバイス、例えばプリンタ102から、管理データ(MIB情報)を取得する。ネットワークデバイス管理プログラム1052は取得した管理データをもとにH

TMLで記述されたページ(HTMLドキュメント)を生成し、Webサーバプログラム1051に返す。Webサーバプログラム1051は、生成されたページをWebクライアントのブラウザプログラム1031に送信する。ブラウザプログラム1031は、受信したページを、その記述に従って表示し、ユーザにより入力されたデータをWebサーバに送信する。

【0037】なお、このネットワークデバイス管理プログラムには、クライアントのWebブラウザプログラム1031からのMIB情報表示要求がある毎にネットワークデバイスから管理データを取得するのでは、クライアントが管理データを表示するまでに時間がかかりすぎるため、管理対象デバイスから取得した管理データをメモリ(RAM)やファイル(HD)に保存(以下、保存したデータをキャッシュという)し、新たに管理データを取得してから特定の期間内にあった要求に対しては、管理対象デバイスから新たに管理データをとってくるのではなく、すでにキャッシュに保存されている管理データを表示するように動作するものもある。

【0038】ここでネットワークデバイス管理プログラム1052が生成した、HTMLドキュメントのブラウザプログラムによる表示例を、図7、図8、および図9に示す。

【0039】図7は、ネットワークに接続された管理対象デバイスを一覧表示する、デバイスリストの表示例である。この例では、ネットワークに接続された、各ネットワークデバイスから、デバイス名、製品名、ネットワークインターフェースボード名、ネットワークアドレス、および状態といったMIB情報を取得し、表示している。

【0040】図8は、図7において、特定のデバイスを選択した際に表示される、選択されたデバイスのより詳細な情報を表示するためのデバイス詳細の表示例である。この例では、ネットワークデバイス(プリンタ)の状態、オプションの装着状況、および給紙部の状態などをMIB情報として取得し、表示している。

【0041】図9は、図8に示した特定のデバイスに関する情報設定画面の一例である。この例では、MIB情報の一部として、ネットワークデバイス(プリンタ)の給紙部に関する設定を行うためのWebページを表示している。図9においては、「給紙部」「自動給紙」「トレイ給紙サイズ」「封筒サイズ」の4つの欄は<SELECT>及び<OPTION>タグにより記述されたプルダウン形式の選択欄(Selectオブジェクト)であり、表示されるプルダウンメニュー中から所望の値を選択できる。また、「トレイ優先」欄は<INPUT>タグのTYPE属性としてCheckboxを指定して記述した選択欄(Checkboxオブジェクト)であり、ユーザはチェックボックスを付して選択の有無を指定できる。これらの欄は、<FORM>タグにより定義された設定形式中に記述されている。図9にお

9

いては、おおむね領域901の部分が設定形式に相当する。なお、設定形式中の<INPUT>タグや<SELECT>タグなどで入力可能に記述された欄を設定形式オブジェクトと呼ぶことにする。また、[OK]ボタンは<INPUT>タグのTYPE="submit"属性で記述されており、これがクリックされると、設定形式中の設定形式オブジェクトをWebサーバに送信する。

【0042】このように、従来、図9に示したような設定用のWebページにおいて[送信]、[OK]などの、<INPUT TYPE="Submit">なるHTMLタグにより記述されたボタン、あるいはそれに準ずる効果をもつボタンを押下した場合、Webページ内に有る、ユーザが入力可能なオブジェクト（INPUTタグのTYPE属性で指定されるCheckboxオブジェクト、Radioオブジェクト、Selectオブジェクト、Textオブジェクトなど）の設定内容が、CGIインタフェースを介してWebサーバに送信される。

【0043】これら情報を受信したWebサーバは、受信した設定値を、SNMPプロトコルを用いて対応するデバイスの対応するMIBオブジェクトに設定していた。

【0044】

【発明が解決しようとする課題】しかしながら上記従来例では、Webサーバ上で動作するCGIアプリケーション（CGIインターフェースを実現するためのプログラム）が常駐型でない、あるいはCGIアプリケーションがデバイスから取得したMIBオブジェクトの値を一時的に保持しておくためのメモリやハードディスクを確保できないなどの理由により、デバイスから取得したMIB情報を一時的に保存するためのキャッシュを利用できない場合、デバイスのMIB情報を変更するためにWebブラウザ上に表示された設定用のWebページに対してユーザが入力したMIB情報の値をCGIインタフェースを介して受信した際に、その受信したMIB情報の値が更新されているかどうかを判断することができない。このため、値が更新されたか否かにかかわらず受信したすべてのMIB情報の値をデバイスに送信しなければならなくなってしまう、ネットワークデバイス管理プログラムとデバイスとの間のトラフィックが増加してしまったり、デバイスの負荷が増加してしまうという問題があった。

【0045】また、設定形式オブジェクトのうち、チェックされていない状態のCheckboxオブジェクトおよびRadioオブジェクト、選択されていない状態のSelectオブジェクトなどは、オブジェクトに関連する情報がCGIパラメータ（CGIに渡される値）に一切載らないため、Webサーバ側のネットワークデバイス管理プログラムで現在の設定要求に対応する設定ページのオブジェクト構成を管理していない限り、CGIパラメータに反映されないMIB情報については、デバイスのMIB情

10

報（MIBオブジェクト）の値を変更することができないという問題があった。この問題に対応するため、ネットワークデバイス管理プログラム側で、設定ページ中に存在する設定形式オブジェクトの構成を管理することもできる。しかしながらこの場合、MIB情報の設定のために生成されるHTMLドキュメントに対して、設定形式オブジェクトの変更を伴う変更があると、ネットワークデバイス管理プログラムで管理しているオブジェクトも変更しなければならず、変更の手間が煩雑になっていまいという問題点があった。

【0046】本発明の目的は上記問題を解決することにより、デバイスのMIB情報の保守を行う場合、ユーザによる変更が可能なMIB情報については、ひとつのデータについて、ユーザによる変更が不可な非表示形式（隠し形式）と、ユーザによる変更が可能な表示形式（設定形式）の2つの形式でクライアントに送り、隠し形式でクライアントに送ったMIB情報に基づいて、値が変更されたMIB情報だけをデバイスに送信することができ、ネットワークデバイス管理プログラムとデバイスとの間のトラフィックの増加や、デバイスの負荷の増加を防止するとともに、ネットワークデバイス管理プログラム側で、設定ページ中に存在する設定形式オブジェクトの構成を管理する必要をなくすることで、HTMLドキュメントあるいはそのテンプレート等の変更が容易に行えるネットワーク管理装置及び方法を提供することを目的とする。

【0047】

【課題を解決するための手段】上記目的を達成するために本発明は次のような構成からなる。すなわち、ネットワークに接続されたネットワークデバイスを管理するネットワーク管理方法であって、デバイスからそのデバイスの設定情報を取得する情報取得工程と、前記設定情報を設定し直すための文書データを、前記設定情報を書き換え不可能な形式で含むように作成する作成工程と前記作成工程において作成された文書データに基づいて設定し直された設定情報を含むデータを受信し、前記書き換え不可能な形式で含ませたデータと比較する比較工程と、前記比較工程による比較の結果、設定し直された設定情報に限って前記デバイスに対して設定する設定工程とを備える。

【0048】また好ましくは、前記文書データはHTML形式の文書データであり、前記ネットワークに接続された端末において表示及び設定し直しが行われる。

【0049】また好ましくは、サーバから読み込んだ情報をキャッシュするキャッシュ機能を有する端末において、サーバからの情報を変更して再表示させるネットワーク管理方法であって、前記端末に送信する情報に含まれる、該情報アクセスするためのリンク情報に、送信する毎に固有な情報を付加する工程と、前記固有な情報が付加されたリンク情報を含む情報を、前記端末に送信す

る工程とを備える。

【0050】また好ましくは、前記付加工程は、HTML文書データに含まれるリンク情報として、送信する毎に固有な情報を付加する。

【0051】また好ましくは、前記端末に送信する情報は、前記端末により設定される、ネットワークデバイスの設定情報である。

【0052】また好ましくは、前記送信する毎に固有な情報は、時刻を示す情報である。

【0053】また好ましくは、前記送信する毎に固有な情報は、所定数を法として漸増する数である。

【0054】

【発明の実施の形態】【第1の実施の形態】以下、図面を用いて本発明の実施例であるネットワーク管理方法について説明する。

【0055】本発明にかかるネットワーク管理方法または装置は、図3に示すような、従来のネットワーク管理装置を実現可能なPCと同様の構成のPC上に実現される。

【0056】図3において、150は、ネットワークデバイス制御ソフトウェアが稼動するPCであり、図1におけるPC150と同等である。PC150は、ROM302もしくはハードディスク(HD)311に記憶された、あるいはフロッピーディスク(FD)312より供給されるネットワークデバイス管理プログラムを実行するCPU301を備え、システムバス304に接続される各デバイスを総括的に制御する。

【0057】303はRAMで、CPU301の主メモリ、ワークエリア等として機能する。305はキーボードコントローラ(KBC)で、キーボード(KB)309や不図示のポインティングデバイスなどからの指示入力を制御する。306はCRTコントローラ(CRTC)で、CRTディスプレイ(CRT)310の表示を制御する。307はディスクコントローラ(DKC)で、ブートプログラム、種々のアプリケーション、編集ファイル、ユーザファイルそしてネットワークデバイス管理プログラム等を記憶するハードディスク(HD)311およびフロッピーディスク(FD)312とのアクセスを制御する。308はネットワークインタフェースカード(NIC)で、LAN100を介して、エージェンツあるいはネットワーク機器と双方向にデータをやりとりする。

【0058】尚、後述のすべての説明において、特に断りのない限り、実行のハード上の主体はCPU301であり、ソフトウェア上の主体はハードディスク(HD)311にインストールされたネットワークデバイス管理ソフトウェアである。

【0059】また、図1において、ネットワークに接続されたネットワークボード(NB)101とそのネットワークボードが装着されたプリンタ102のような組み

合わせをネットワークデバイスと呼ぶ。

【0060】図4は、本発明に係るネットワーク管理ソフトウェア(プログラム)のモジュール構成図である。

【0061】本発明に係るネットワーク管理ソフトウェア1052は、図3におけるハードディスク(HD)311に格納されており、CPU301によって実行される。その際、CPU301はワークエリアとしてRAM303を使用する。

【0062】図4において、ネットワーク管理ソフトウェア1052は、Webサーバプログラム1051から起動され、CGIインタフェース402を介してCGIパラメータ、およびHTMLドキュメントのやり取りを行う。

【0063】403は全体制御モジュールで、後述のパラメータモジュール404に対してCGIパラメータを登録したのち、CGIパラメータの中のコマンドパラメータに応じて、後述のシステムモジュール405、デバイスリストモジュール407、デバイス詳細モジュール409のいずれかに制御を振り分けるためのモジュールである。CGIパラメータに誤りがあった場合は、後述のテンプレートモジュール412を介して、CGIパラメータに誤りがある旨のHTMLドキュメントを生成することもある。

【0064】404はパラメータモジュールで、全体制御モジュール403により登録されたCGIパラメータをテーブル形式で保存・管理するモジュールである。他のモジュールは、必要に応じて、パラメータモジュール402から、所望するパラメータを取得することができる。

【0065】405はシステムモジュールで、ネットワーク管理ソフトウェア1052の動作を規定するシステムパラメータ(例えば、HTMLドキュメントの自動更新間隔など)の表示・設定を制御し、また、関連するHTMLドキュメントを生成するモジュールである。システムモジュール405は、パラメータモジュール404からコマンドパラメータを取得し、コマンドパラメータの内容がシステムパラメータの表示要求であった場合は、システム設定ファイル406から必要な情報を読み出し、テンプレートモジュール412を介してシステムパラメータ表示用のHTMLドキュメントを生成する。また、取得したコマンドパラメータの内容がシステムパラメータの設定要求であった場合は、システム設定ファイル406に対して通知されたシステムパラメータを書き込み、テンプレートモジュール412を介して、設定後に表示すべきHTMLドキュメントを生成する。尚、図には示していないが、システム設定ファイル406に保存したシステムパラメータは、ネットワーク管理ソフトウェア1052を構成する各モジュールが、必要に応じて読み出すことができる。

【0066】407はデバイスリストモジュールで、後



述のデバイス探索モジュール408により探索されたデバイスの一覧（デバイスリスト）を示すHTMLドキュメントを生成するためのモジュールである。デバイスリストの表示オプションの処理なども、このデバイスリストモジュール407が制御する。

【0067】408は、デバイス探索モジュールで、ネットワークに接続されたデバイスを探索するモジュールである。

【0068】409はデバイス詳細モジュールで、CGIパラメータにより指定された特定のデバイスに対するより詳細な情報を表示・設定するための制御、および関連するHTMLドキュメントを生成するためのモジュールである。デバイス詳細モジュール409は、指定されたデバイスの詳細な情報を取得・設定するために、指定されたデバイスに対応する後述のデバイス固有モジュール410を使用する。

【0069】410はデバイス固有モジュールで、ネットワーク管理ソフトウェアの管理対象となるデバイス（プリンタ、ネットワークインタフェースボードなど）ごとに存在する。デバイス固有モジュール410は、表示時においては、デバイスから必要な情報を取得し、取得した情報をテンプレートモジュール412に設定するように動作する。また、設定時においては、CGIパラメータで通知された設定値を、デバイスが解釈可能な値に変換し、デバイスに送信するように動作する。

【0070】411はプロトコルモジュールで、MIBのハンドリング、SNMPパケットの送受信、トランスポートプロトコルの制御など、ネットワーク管理ソフトウェアがデバイスと通信するために必要な、各種プロトコルの制御を行うためのモジュールである。

【0071】412はテンプレートモジュールで、図3のハードディスク311に保存されているテンプレートファイル413を元に、ネットワーク管理ソフトウェアの出力結果としてのHTMLドキュメントを生成するためのモジュールである。テンプレートモジュール412は、CGIパラメータ、全体制御モジュール403、システムモジュール405、デバイスリストモジュール407、またはデバイス詳細モジュール409により指定されたテンプレートファイルをオープンし、テンプレートファイルの内容を解析し、必要に応じてテンプレートファイルに含まれるテンプレート変数を、全体制御モジュール403、システムモジュール405、デバイスリストモジュール407、デバイス詳細モジュール409、またはデバイス固有モジュール410により設定された値に置き換えることにより、HTMLドキュメントを生成し、CGIインタフェース402を介してWebサーバプログラムに送信する。HTMLドキュメントを生成する際に使用したテンプレート変数の値、あるいは、生成したHTMLドキュメントファイルは、同一のテンプレートファイルを元にして2回目以降にHTML

ドキュメント生成する際の処理時間を短縮するために、キャッシュファイル414として図3のハードディスク311上に保存することもできる。

【0072】図5は、本発明のネットワークデバイス管理ソフトウェアにおいて、WebブラウザからのCGIリクエストに応じて、HTMLドキュメントを生成する処理の一例を示すフローチャートである。

【0073】同図において、ステップS501では、管理対象デバイスに対してGet-request、Get-next-requestコマンドを送信し、Get-responseコマンドを受信することにより、ネットワークを経由して、要求されたHTMLドキュメントを生成するために必要なMIB情報を、当該管理対象デバイスから取得する。尚、図には示していないが、Get-request、Get-next-requestに対して期待されるGet-responseが受信できなかった場合は、コマンドの再送などの適切なエラー回復処理が起動される。また、管理対象デバイスは、通常、WebブラウザからCGIパラメータの1つとして、ネットワークアドレスのような形で指定される。

【0074】ステップS502では、本CGIリクエストのCGIパラメータの1つとして指定されたテンプレートファイル（Webサーバ上に保存されている）をオープンし、ファイルの内容（テンプレート）を作業用メモリに読み込み、テンプレートファイルをクローズした後、ステップS503に進む。

【0075】ステップS503では、前記ステップS502で読み出したテンプレートから、トークン（テンプレートの解析単位）を1つ取り出し、ステップS504へ進む。

【0076】ステップS504では、前記ステップS503でのトークン取得処理が成功したかどうか調べ、トークンの取得に成功している場合はステップS505に進み（ステップS504：Yes）、失敗している場合はステップS511に進む（ステップS504：No）。尚、トークンの取得失敗は、テンプレート中のすべてのトークンの解析が終了したことを意味する。

【0077】ステップS505では、前記ステップS503で取得したトークンを解析し、その種別を調べ、ステップS506に進む。

【0078】ステップS506では、前記ステップS505での解析の結果、トークンがデバイスから取得したMIB情報の埋め込み指示である場合はステップS507に進み（ステップS506：Yes）、それ以外の場合はステップS510に進む（ステップS506：No）。

【0079】ステップS507では、前記MIB情報の埋め込み指示の対象となるMIB情報（デバイスレベルでは、MIBオブジェクト）が、ユーザによる閲覧のみ

が可能な情報（デバイスレベルでは、リードオンリーのMIBオブジェクト）であるか、ユーザによる変更および設定が可能な情報（デバイスレベルでは、リードライト可能なMIBオブジェクト）であるかを判別し、ユーザによる閲覧のみが可能な情報である場合はステップS508に進み（ステップS507：No）、ユーザによる変更および設定が可能なオブジェクトの場合はステップS509に進む（ステップS507：Yes）。

【0080】ステップS508では、HTMLドキュメントに、前記トークンにより指定されたMIB情報の値<sup>10</sup>を閲覧形式オブジェクトとして埋め込み、ステップS503に進む。尚、埋め込むMIB情報を識別するために、MIB情報の名前などの付加情報が必要な場合は、前記トークンに引き続く情報を、前記テンプレートから読み出すこともある。また、ここで言う閲覧形式オブジェクトとは、HTMLの<INPUT>タグなどを使わないプレーンな文字列（<FONT COLOR>タグなどによる装飾などはあってもよいものとする）を意味する。

【0081】ステップS509では、HTMLドキュメントに、前記トークンにより指定されたMIB情報の値<sup>20</sup>を隠し形式オブジェクトと設定形式オブジェクトの2つの形式で埋め込み、ステップS503に進む。尚、埋め込むMIB情報を識別するために、MIB情報の名前などの付加情報が必要な場合は、前記トークンに引き続く情報を、前記テンプレートから読み出すこともある。また、ここで言う隠し形式オブジェクトとは、HTMLのHiddenオブジェクト（<INPUT TYPE=“hidden”>タグを使って記述）を意味し、設定形式オブジェクトとは、HTMLのCheckboxオブジェクト（<INPUT TYPE=“checkbox”>タグを使って記述）、Radioオブジェクト（<<sup>30</sup>INPUT TYPE=“radio”>タグを使って記述）、Selectオブジェクト（<SELECT>タグと<OPTION>タグを使って記述）、Textオブジェクト（<INPUT TYPE=“text”>タグを使って記述）、Textareaオブジェクト（<TEXTAREA>タグを使って記述）などに代表されるユーザからの入力可能な表示形式を意味する。

【0082】ステップS510では、MIB情報の埋め込み指示以外のトークンに関する処理を行い、ステップS503に進む。ここでの処理は、例えば、テンプレート中の記述をそのままHTMLドキュメントに埋め込む<sup>40</sup>などの処理があげられるが、詳細については本発明の範囲外であるので省略する。

【0083】ステップS511では、前記ステップまでに生成したHTMLドキュメントをWebブラウザに送信するために、標準出力に出力し、作業用メモリの解放などの必要な後処理を行った後に本処理を終了する。

【0084】以上の手順によって、デバイスから読み出したMIB情報を埋め込んだ、クライアント端末においてブラウザプログラムにより表示させるためのHTML文書を作成する。

【0085】ここで、本願のネットワークデバイス管理ソフトウェアが図5の手順により生成したHTMLドキュメント（設定用のWebページ）の例を図10に示す。図10の記述により、従来例として図9に示した設定型式欄901と同一の表示がブラウザプログラムによりなされる。ただし、HTML文書の内容は図10に示すとおり従来の文書とは異なっている。

【0086】図10に示したHTMLドキュメントの場合、2行目から7行目までが隠し形式オブジェクトとして埋め込まれたMIB情報であり、9行目から71行目までのテーブルの中で設定可能オブジェクトとして前記隠し形式オブジェクトに対応するMIB情報が埋め込まれている。この例では、対応するMIB情報は、HTML中に埋め込まれたオブジェクトの名前で識別できるようになっており、設定形式オブジェクトの名前の先頭にアンダースコア“\_”を付加すると隠し形式オブジェクトの名前になるようになっている。例えば、隠し形式オブジェクト（Hiddenオブジェクト）の\_InputSelected（2行目）は、設定形式オブジェクト（Selectオブジェクト）のInputSelected（22行目）に対応している。これら設定可能オブジェクトは、隠し形式として埋め込まれた値が初期的に選択された状態で表示されるよう、属性が設定されている。

【0087】クライアントのブラウザプログラムは、図10のHTML文書を受け取ると、それを図9のように表示し、ユーザに所定のオブジェクトを設定させる。設定が終わり、ユーザが[OK]ボタンを押すと、設定型式欄のうち、選択された値、及び、チェックされた値がサーバのネットワークデバイス管理ソフトウェアに返される。

【0088】図6は、本発明のネットワークデバイス管理ソフトウェアにおいて、CGIリクエストによるデバイスへのMIB情報設定処理の一例を示すフローチャートである。

【0089】同図において、ステップS601では、前記図5の処理によりHTMLドキュメント生成時に埋め込まれた、デバイスから取得したMIB情報の値（隠し形式オブジェクトとして埋め込まれている）と、Webブラウザ上でユーザが入力したMIB情報の値（設定形式オブジェクトに対して入力された値）を比較するための表を作成し、ステップS602に進む。

【0090】表1に、図10に示したHTMLドキュメントに対して、ユーザがMIB情報の変更を行った場合に通知されるCGIパラメータ（一部抜粋）と、そのCGIパラメータからステップS601で作成される、隠し形式オブジェクトの値（\_InputSelected=1, \_InputAutoFeed=1, \_InputAutoFeed=2, \_InputAutoFeed=3, \_PreferenceTray=1, \_TrayPaperSize=3, \_EnvelopeSize=1）と、設定形式オブジェクトとして入力された値（本例では、InputSelected=1, InputAutoFeed=3, InputAutoFee

d=4, TrayPaperSize=1, EnvelopeSize=1) の一例との対 \*【0091】  
 応表を示す。 \* 【表1】

	オブジェクト名 (MIB 情報名)	隠し形式オブジェクトの値	設定形式オブジェクトの値
0	InputSelected	1	1
1	InputAutoFeed	1	—
2	InputAutoFeed	3	3
3	InputAutoFeed	—	4
4	PreferenceTray	1	—
5	TrayPaperSize	3	1
6	EnvelopeSize	1	1

【0092】表中、設定形式オブジェクトの値に“—”と記してあるものは、オブジェクトが非チェック状態 (Checkbox オブジェクト、Radio オブジェクトの場合) または非選択状態 (Select オブジェクトの場合) になったことを意味している。また、隠し形式オブジェクトの値が“—”と記してあるものは、デバイスから取得した 20 値は非チェック状態 (Checkbox オブジェクト、Radio オブジェクトの場合) または非選択状態 (Select オブジェクトの場合) のオブジェクトに相当する状態であったが、ユーザが設定形式オブジェクトで値を変更したことによりオブジェクトの値が変更されたことを示している。また、隠しオブジェクトの値も、設定形式オブジェクトの値も、共に、非チェック状態 (Checkbox オブジェクト、Radio オブジェクトの場合) または非選択状態 (Select オブジェクトの場合) の場合 (すなわち値が変更されない場合) は、そのオブジェクトに関する情報はこ 30 の表中には現れない。例えば、InputAutoFeed の 2 番目の要素 (値が 2 の要素、図 9 で言えば「上段カセット」) は、デバイスから取得した情報が非選択状態であり、ユーザが入力した結果も非選択状態であったため、隠し形式でも設定形式でも値は存在しないため、前記表中には現れない。

【0093】ステップ S602 では、変数 n に、前記ステップ S601 で作成した対応表のサイズ (行数) を代入し、ステップ S603 に進む。前記ステップ S601 の説明で示した図 10 の例では、対応表のサイズは 40 “7” となる。

【0094】ステップ S603 では、変数 i を“0”に初期化し、ステップ S604 に進む。

【0095】ステップ S604 では、前記変数 i と前記変数 n の値を比較し、変数 i の値が変数 n の値よりも小さい場合はステップ S605 に進み (ステップ S604 : Yes)、それ以外の場合はステップ S608 に進む (ステップ S604 : No)。

【0096】ステップ S605 では、前記ステップ S601 で作成した対応表の i 番目の隠しオブジェクトの値 50

と設定形式オブジェクトの値を比較し、値が異なる場合はステップ S606 に進み (ステップ S605 : No)、値が一致する場合はステップ S607 に進む (ステップ S605 : Yes)。

【0097】ステップ S606 では、前記ステップ S601 で作成した対応表の i 番目の設定形式オブジェクトの値を、当該設定に対応するデバイスの当該情報に対応する MIB オブジェクトに対して SNMP プロトコルの Set-request コマンドを用いて送信し、ステップ S607 に進む。尚、当該オブジェクトの設定において、タイムアウトなどが発生した場合は、所定のリトライ処理などを行う。例えば、前記ステップ S601 の説明で示した、対応表の一例 (表 1) を例にとると、

1、3、4、5 行目のオブジェクトの値が、HTML ドキュメント生成時に埋め込まれた隠しオブジェクトの値と、Web ブラウザ上でユーザが設定形式オブジェクトに対して入力した値とで異なっているため、当該 MIB 情報の値を対応するデバイスに対して送信する。尚、MIB 情報値の送信に際しては、MIB 情報の名前から MIB オブジェクトのオブジェクト識別子への変換や、MIB 情報の値から MIB オブジェクトに設定すべき適切なデータ形式への変換が必要に応じて行われる。

【0098】ステップ S607 では、前記変数 i の値に“1”を加算し、ステップ S604 に進む。

【0099】ステップ S608 では、前記ステップまでの処理に引き続き、CGI パラメータで指定されたテンプレート名などを元に、前記図 5 に示した手順に従って、Web ブラウザに次に表示する Web ページのための HTML ドキュメントを生成し、処理を終了する。

【0100】また、上記で説明した本発明に係るネットワークデバイス制御ソフトウェアは、外部からインストールされるプログラムによって、ネットワークデバイス制御ソフトウェアが稼動可能な PC の構成を示すブロック図である図 3 における PC150 と同等の構成をもつ PC によって遂行されても良い。その場合、そのプログラムは CD-ROM やフラッシュメモリやフロッピーデ

ディスクなどの記憶媒体により、あるいは電子メールやパソコン通信などのネットワークを介して、外部の記憶媒体からプログラムを含む情報群をPC上にロードすることにより、PCに供給される場合でも本発明は適用されるものである。

【0101】図11は、記憶媒体の一例であるCD-ROMのメモリマップを示す図である。9999はディレクトリ情報を記憶してある領域で、以降のインストールプログラムを記憶してある領域9998およびネットワークデバイス制御ソフトウェアを記憶してある領域9997の位置を示している。9998は、インストールプログラムを記憶してある領域である。9997は、ネットワークデバイス制御ソフトウェアを記憶してある領域である。本発明のネットワークデバイス制御ソフトウェアが、図3に示したPC150と同等の構成をもつPCにインストールされる際には、まずインストールプログラムを記憶してある領域9998に記憶されているインストールプログラムがシステムにロードされ、CPU301によって実行される。次に、CPU301によって実行されるインストールプログラムが、ネットワークデバイス制御ソフトウェアを記憶してある領域9997からネットワークデバイス制御ソフトウェアを読み出して、ハードディスク311に格納する。

【0102】以上説明したように、本実施例のネットワーク管理装置および方法によれば、デバイスからMIB情報を取得し、デバイスにMIB情報を設定し、Webブラウザからの表示リクエストがあった場合に、HTMLドキュメントを生成し、前記HTMLドキュメント中にMIB情報を埋め込む場合に、前記MIB情報が閲覧のみ可能な情報の場合は、ユーザによる変更が不可な表示形式のオブジェクト（閲覧形式オブジェクト）として、前記MIB情報がユーザによる変更が可能な情報の場合は、ユーザによる変更が不可な非表示形式のオブジェクト（隠し形式オブジェクト）と、ユーザによる変更が可能な表示形式のオブジェクト（設定形式オブジェクト）の2つの形式で埋め込むように制御し、Webブラウザからの設定リクエストがあった場合に、当該リクエストのCGIパラメータとして通知された、前記生成したHTMLドキュメントに対してユーザがWebブラウザ上で設定形式オブジェクトに対して設定した値と、当該HTMLドキュメントを生成する際に埋め込んだ隠し形式オブジェクトの値とを比較し、値が変更されたMIB情報のみをデバイスに送信するように制御する。こうすることにより、ネットワークデバイス管理プログラムが非常駐型であったり、ネットワーク管理装置または方法がデバイスから取得したMIB情報を一時的に保存するためのメモリやハードディスクを確保することができない場合であっても、WebブラウザからのCGIパラメータで通知された情報のみを解析することで、必要十分なMIB情報を対応するデバイスに送信することが可

能になるという効果がある。

【0103】また、ネットワークデバイス管理プログラム側で、設定ページ中に存在する設定形式オブジェクトの構成を管理する必要がなくなるため、HTMLドキュメントを生成するためのテンプレートの変更などが容易になるという効果がある。

【0104】〔第2の実施の形態〕本実施形態は、第1の実施形態と同様のネットワークシステムについて説明する。

【0105】第1の実施形態におけるWWWシステムを利用したでは、Webブラウザプログラム1031は、デバイスの管理情報を変更する都度、サーバのネットワークデバイス管理プログラムによりネットワークデバイスから取得したMIB情報をHTML文書として受信し、それを表示している。しかしながら、今日普及している多くのWebブラウザは、Webサーバから取得したHTMLドキュメントを、そのドキュメントを取得したときのロケーション情報（URL）と共にブラウザのキャッシュとしてローカル端末上のハードディスクまたはメモリに保存し、それ以降に同一のURLに対するリクエストがあった場合に、Webサーバから情報を取得するのではなく、ブラウザのキャッシュに保存したHTMLドキュメントを表示することにより、リクエストが発行されてからWebページが表示までにかかる時間を短くする機能をもっている。このようなWebブラウザでは、ブラウザのキャッシュ機能を使うかどうかを、Webブラウザのオプション設定で変更することが可能なものが多い。

【0106】しかしながら、Webブラウザのオプション設定で、ブラウザのキャッシュを使用するように設定した場合、URLにCGIリクエストを指定した場合もブラウザのキャッシュに保存されたHTMLドキュメントが表示されてしまう。このため、例えば、CGIリクエストを実行するたびにその結果として返されるWebページの内容（HTMLドキュメント）が変化する場合に、その内容が表示に反映されなくなってしまう。

【0107】具体的には、

（1）WebブラウザからのCGIリクエストにより、デバイスの情報を閲覧するWebページを表示する。

【0108】（2）前記情報閲覧用のWebページに用意された「変更」をクリックすることにより、図9に示すようなデバイスの情報を設定するWebページに移動する。

【0109】（3）前記情報設定用のWebページでデバイスの情報を変更し、更新内容をCGIパラメータでWebサーバ側に通知するために「OK」ボタン（または、それに準ずるボタン）を押す。といった操作をした場合に、（3）で「OK」を押した後に、前記（1）のCGIリクエストで指定したURLと同一のWebページ

ジを表示するように実装されている場合、ブラウザのキャッシュ機能が使用されると、前記(1)でブラウザのキャッシュに保存されたHTMLドキュメントが表示されてしまう。そのため、(3)での情報の変更がWebページに反映されなくなってしまう。

【0110】この問題は、Webブラウザのキャッシュ機能を無効にすることにより回避することは可能であるが、そうした場合、一度取得してしまえばそれ以降は内容が変化しないようなWebページを表示する際にも、そのつどWebサーバからHTMLドキュメントを取得することになってしまうため、Webブラウザ・Webサーバ間のトラフィックが増えたり、表示が更新されるまでに時間がかかってしまうためにユーザの操作性が悪くなってしまう。

【0111】本実施形態では、このようにブラウザのキャッシュ機能を利用する場合であっても、設定系のWebページで値を変更した後、表示系のWebページで変更した値を閲覧するような場合に、ブラウザのキャッシュに保存されている古いHTMLドキュメントを表示させず、設定系のWebページで変更された最新の値を表示できるネットワーク管理装置及び方法を説明する。

【0112】図12は、本実施形態におけるネットワークデバイス管理プログラム1052による、クライアントからCGIリクエストを受信した際の手順である。

【0113】図12において、ステップS1201では、管理対象デバイスに対してGet-request、Get-next-requestコマンドを送信し、それに対するGet-responseコマンドを受信することにより、ネットワークを経由して、当該管理対象デバイスの管理データを取得し、ステップS1202に進む。尚、図には示していないが、Get-request、Get-next-requestに対して期待されるGet-responseが受信できなかった場合は、コマンドの再送などの適切なエラー回復処理が起動される。また、管理対象デバイスは、通常、WebブラウザからCGIパラメータの1つとして、ネットワークアドレスのような形で指定される。更に、本CGIパラメータには、現在表示されているHTMLドキュメントを生成する際に埋め込まれた時刻パラメータが付加されている場合があるが、そのパラメータは本処理の過程では使用されない(無視する)。

【0114】ステップS1202では、本CGIリクエストのCGIパラメータの1つとして指定されたテンプレートファイル(Webサーバ上に保存されている)をオープンし、ファイルの内容(テンプレート)を作業用メモリに読み込み、テンプレートファイルをクローズした後、ステップS1203に進む。

【0115】ステップS1203では、前記ステップS1202で読み出したテンプレートから、トークン(テンプレートの解析単位)を1つ取り出し、ステップS1

204へ進む。

【0116】ステップS1204では、前記ステップS1203でのトークン取得処理が成功したかどうか調べ、トークンの取得に成功している場合はステップS1205に進み(ステップS1204:Yes)、失敗している場合はステップS1211に進む(ステップS1204:No)。尚、トークンの取得失敗は、テンプレート中のすべてのトークンの解析が終了したことを意味する。

【0117】ステップS1205では、前記ステップS1203で取得したトークンを解析し、その種別を調べ、ステップS1206に進む。

【0118】ステップS1206では、前記ステップS1205での解析の結果、取得したトークンが、デバイスから取得したMIB情報の埋め込み指示である場合はステップS1207に進み(ステップS1206:Yes)、それ以外の場合はステップS1208に進む(ステップS1206:No)。

【0119】ステップS1207では、HTMLドキュメントに、前記トークンにより指定されたMIB情報の値を埋め込み、ステップS1203に進む。尚、埋め込むMIB情報を識別するために、MIB情報の名前などの付加情報が必要な場合は、前記トークンに引き続く情報を、前記テンプレートから読み出すこともある。

【0120】ステップS1208では、前記ステップS1205での解析の結果、取得したトークンが、CGIリクエストに対してダミーパラメータを付加するためのスクリプトの埋め込み指示である場合はステップS1209に進み(ステップS1208:Yes)、それ以外の場合はステップS1210に進む(ステップS1208:No)。

【0121】ステップS1209では、HTMLドキュメントに、CGIリクエストを実行したときの時刻情報をCGIリクエストのパラメータの1つ(ダミーパラメータ)として付加するためのスクリプトを埋め込み、ステップS1203に進む。尚、ダミーパラメータを付加するためのスクリプトは、前記のようにスクリプトの埋め込み指としてではなく、後述のステップS1210に示すその他のトークンの処理の一部として、テンプレートからHTMLドキュメントにそのまま複写される文字列として記述してあってもよい。また、前記スクリプトにより付加されるダミーパラメータ(時刻情報)は、Webブラウザにおいてキャッシュに保存されたHTMLドキュメントに対応するURLと、新しいCGIリクエストのためのURLが一致しないようにするためのパラメータであり、Webサーバ側のネットワークデバイス管理プログラムでは無視される。したがって、パラメータの名前は、ネットワークデバイス管理ソフトウェアでリザーブされている他の有効なCGIパラメータの名前と異なっていればなんでもよい。例えば、本例では、

23

“rid”という名前を使っている。

【0122】ステップS1210では、前記ステップS1207およびステップS1209で処理したトークン以外の、他のトークンに関する処理を行い、ステップS1203に進む。ここでの処理は、例えば、テンプレート中の記述をそのままHTMLドキュメントに埋め込むなどの処理があげられるが、詳細については本発明の範囲外であるので省略する。

【0123】ステップS1211では、前記ステップまでに生成したHTMLドキュメントをWebブラウザに送信するために、標準出力に出力し、作業用メモリの解放などの必要な後処理を行った後に本処理を終了する。

【0124】図13は、本発明のネットワークデバイス管理ソフトウェアにおけるHTMLドキュメント生成手段が、HTMLドキュメントを生成する際に、CGIリクエストに対して時刻情報をダミーパラメータとして埋め込むスクリプトの一例である。

【0125】同図に示したスクリプトはJavaScriptで記述されており、getCgi関数は、Webサーバ側のネットワークデバイス管理プログラムがCGIリクエストを発行する際に、管理対象デバイスのネットワークアドレスと、HTMLドキュメントを生成する際に使用するテンプレート名をパラメータとして指定すると、ダミーパラメータ（時刻情報）を含むURL（CGIリクエスト）文字列を返すようになっている。

【0126】同図において、4行目では、本CGIリクエストが起動された時刻情報を取得するために、JavaScriptの基本オブジェクトであるDateオブジェクトを生成している。

【0127】7行目では、関数のパラメータで指定されたネットワークアドレスをaddrパラメータとしてCGIリクエストに付加している。

【0128】8行目では、関数のパラメータで指定されたテンプレート名をtmplパラメータとしてCGIリクエストに付加している。

【0129】9行目では、本CGIリクエストが起動された時刻情報として、4行目で生成したDateオブジェクトのgetTimeメソッドを使って現在の時刻情報（1970年1月1日0時0分0秒からのミリ秒単位の経過時間）を取得し、ridパラメータとしてCGIリクエストに付加している。

【0130】図14は、図13で説明したgetCgi関数の使用例である。

【0131】同図において、getCgi関数を呼び出す側のgotoDeviceDetail関数はJavaScriptで記述されており、<A HREF=“javascript:関数名”>タグなどにより記述されたリンクに関連付けられて起動される。gotoDeviceDetail関数は、図8に示したデバイス詳細ページを表示するためのWebページヘジャンプするための関数であり、パラメータとして管理対象デバイスのネットワー

24

クアドレス（例えば、IPアドレス）を受け取る。

【0132】2行目では、gotoDeviceDetail関数のパラメータで指定されたネットワークアドレスと、デバイス詳細ページを生成するために使用するデフォルトのテンプレート名である“default”をパラメータとして指定してgetCgi関数を呼び出し、getCgi関数の戻り値（CGIリクエスト文字列）をtop.locationに代入することにより、所望のデバイス詳細ページヘジャンプするようにしている。

【0133】このようにして、gotoDeviceDetail関数で与えられるURLを持つWebページをWebブラウザでは読み出す。gotoDeviceDetail関数で与えられるURLは、その関数が実行された時点における時刻の情報を含むため、キャッシュされたURLと一致することではなく、Webサーバに対してあらためてCGIリクエストが送られる。Webサーバのネットワークデバイス管理プログラムは、指定されたURLのうち、ダミーパラメータである時刻情報を無視して、再びデバイスからMIB情報を読み出し、HTML文書を作成してWebクライアントのブラウザに対して送信する。

【0134】このようにして、ブラウザプログラムは例えば図8のページを表示できる。図8の画面のアドレス欄には、gotoDeviceDetail関数で得られるURLである、“wnetspot.exe&addr=\*\*\*.\*\*\*.\*\*\*.\*\*\*&tmpl=default&rid=100”（図8ではgetTimeメソッドにより値100が得られている）が表示されており、関数gotoDeviceDetailにより得られたURLを有するWebページであることがわかる。

【0135】以上のように、クライアントのWebブラウザからのCGIリクエストに応じて、ネットワークデバイス管理プログラムによりネットワークデバイスからMIB情報を取得し、前記デバイスから取得したMIB情報と所定のテンプレートとを元にHTMLドキュメントを生成する際、HTMLドキュメント中にCGIリクエストのパラメータの1つとして、少なくとも前回実行された同一のCGIリクエストと当該CGIリクエストを識別することが可能なダミーパラメータ（本例では時刻を含むURL）を埋め込む。こうすることにより、Webブラウザは、そのキャッシュ機能が有効な場合であっても、HTML文書に埋め込まれたダミーパラメータを含むURLのページを表示するため、Webブラウザのキャッシュに保存されたHTMLドキュメントは表示されることはない。

【0136】これにより、例えば、設定系のWebページで値を変更した後、表示系のWebページで前記変更した値を閲覧するような場合に、表示系のWebページを表示する際にブラウザのキャッシュに保存されている古いHTMLドキュメントが表示されてしまうことにより、前記設定系のWebページで変更したはずの最新の値と異なる値が表示されてしまうという不都合を回避す

25

ることが可能になる。

【0137】〔第3の実施の形態〕本実施形態は、第2の実施形態と同様のネットワークシステムにおいて、ダミーパラメータとして時刻以外の情報を用いる方法について説明する。

【0138】図15は、本実施形態におけるネットワークデバイス管理プログラム1052による、クライアントからCGIリクエストを受信した際の手順である。

【0139】図15において、ステップS1501では、管理対象デバイスに対してGet-request 10、Get-next-requestコマンドを送信し、Get-responseコマンドを受信することにより、ネットワークを経由して、当該管理対象デバイスの管理データを取得し、ステップS1502に進む。尚、図には示していないが、Get-request、Get-next-requestに対して期待されるGet-responseが受信できなかった場合は、コマンドの再送などの適切なエラー回復処理が起動される。また、管理対象デバイスは、通常、WebブラウザからCGIパラメータの1つとして、ネットワークアド 20レスのような形で指定される。更に、本CGIパラメータには、現在表示されているHTMLドキュメントを生成する際に埋め込まれたダミーパラメータが付加されている場合があるが、そのパラメータは本処理の過程では使用されない（無視する）。

【0140】ステップS1502では、本処理で生成するHTMLドキュメントにおいて、CGIリクエストを実行するためのCGIパラメータに付加するダミーパラメータの値を生成し、ステップS1503に進む。尚、ダミーパラメータの値の生成処理の一例については、図 3016で説明する。

【0141】ステップS1503では、本CGIリクエストのCGIパラメータの1つとして指定されたテンプレートファイル（Webサーバ上に保存されている）をオープンし、ファイルの内容（テンプレート）を作業用メモリに読み込み、テンプレートファイルをクローズした後、ステップS1504に進む。

【0142】ステップS1504では、前記ステップS1503で読み出したテンプレートから、トークン（テンプレートの解析単位）を1つ取り出し、ステップS 40505へ進む。

【0143】ステップS1505では、前記ステップS1504でのトークン取得処理が成功したかどうか調べ、トークンの取得に成功している場合はステップS1506に進み（ステップS1505：Yes）、失敗している場合はステップS1512に進む（ステップS1505：No）。尚、トークンの取得失敗は、テンプレート中のすべてのトークンの解析が終了したことを意味する。

【0144】ステップS1506では、前記ステップS 50

26

1504で取得したトークンを解析し、その種別を調べ、ステップS1507に進む。

【0145】ステップS1507では、前記ステップS1506での解析の結果、トークンがデバイスから取得したMIB情報の埋め込み指示である場合はステップS1508に進み（ステップS1507：Yes）、それ以外の場合はステップS1509に進む（ステップS1507：No）。

【0146】ステップS1508では、HTMLドキュメントに、前記トークンにより指定されたMIB情報の値を埋め込み、ステップS1504に進む。尚、埋め込むMIB情報を識別するために、MIB情報の名前などの付加情報が必要な場合は、前記トークンに引き続き情報を、前記テンプレートから読み出すこともある。

【0147】ステップS1509では、前記ステップS1506での解析の結果、トークンがダミーパラメータの埋め込み指示である場合はステップS1510に進み（ステップS1509：Yes）、それ以外の場合はステップS1511に進む（ステップS1509：N 0）。

【0148】ステップS1510では、HTMLドキュメントに、前記ステップS1502で生成した値をCGIリクエストのパラメータの1つ（ダミーパラメータ）として付加し、ステップS1504に進む。尚、前記付加されるダミーパラメータは、Webブラウザにおいてキャッシュに保存されたHTMLドキュメントに対応するURLと、新しいCGIリクエストのためのURLが一致しないようにするためのパラメータであり、Webサーバ側のネットワークデバイス管理プログラムでは無視される。したがって、パラメータの名前は、ネットワークデバイス管理ソフトウェアでリザーブされている他の有効なCGIパラメータの名前と異なっていればなんでもよい。例えば、後で説明する図17、図18では、“rid”（システム内の変数名としては“\_\_RID\_\_”）という名前を使っている。

【0149】ステップS1511では、前記ステップS1507からステップS1510で処理したトークン以外の、他のトークンに関する処理を行い、ステップS1504に進む。ここでの処理は、例えば、テンプレート中の記述をそのままHTMLドキュメントに埋め込むなどの処理があげられるが、詳細については本発明の範囲外であるので省略する。

【0150】ステップS1512では、前記ステップまでに生成したHTMLドキュメントをWebブラウザに送信するために、標準出力に出力し、作業用メモリの解放などの必要な後処理を行った後に本処理を終了する。

【0151】図16は、本発明のネットワークデバイス管理ソフトウェアにおいて、CGIリクエストに付加するダミーパラメータの値を生成する処理の一例を示すフローチャートである。ダミーパラメータの値の生成方法

27

としては、Webサーバ上におけるシステム時間を利用する方法、ランダム値を生成するなんらかのアルゴリズムを利用する方法など、いろいろな方法が考えられるが、本例では、ダミーパラメータの値として十分大きなモジュロをもつシーケンシャルな値（モジュロ10000）を使用している。

【0152】同図において、ステップS1601では、前回のダミーパラメータの値\_\_RID\_\_をWebサーバ上の所定のファイルから読み出し、ステップS1602に進む。尚、前回のダミーパラメータの値が所定のファイルに保存されていなかった場合の初期値は“0”とする。

【0153】ステップS1602では、前記ステップS1601で取得したダミーパラメータの値\_\_RID\_\_に“1”を加算し、ステップS1603に進む。

【0154】ステップS1603では、ダミーパラメータの値\_\_RID\_\_が“9999”よりも大きいかどうか比較し、大きい場合はステップS1604に進み（ステップS1603：Yes）、大きくない場合はステップS1605に進む（ステップS1603：No）。<sup>20</sup>

【0155】ステップS1604では、ダミーパラメータの値\_\_RID\_\_を“0”に初期化し、ステップS1605に進む。

【0156】ステップS1605では、今回のダミーパラメータの値\_\_RID\_\_を、前記ステップS1601において値\_\_RID\_\_を読み出した所定のファイルに書き込み、ステップS1606に進む。

【0157】ステップS1606では、今回のダミーパラメータの値\_\_RID\_\_を返し、処理を終了する。

【0158】このようにして、ダミーパラメータとして十分に大きな値を法として漸増する計数値を用い、そのダミーパラメータをURLに含むように持たせることで、そのURLと同一のURLがWebブラウザのキャッシュにあるという事態を避けることができる。このため、更新されているはずのページが、更新されていない、古い状態で表示されることを防止できる。

【0159】図17は、本発明のネットワークデバイス管理ソフトウェアにおけるHTMLドキュメント生成手段が、HTMLドキュメントを生成する際に使用するテンプレートの一例である。<sup>40</sup>

【0160】同図に示しているのは、図8に示したデバイス詳細ページのトップレベルのフレーム分割（縦に2分割）を記述しているHTMLドキュメントのテンプレートであり、16行目で左のフレームに表示するHTMLドキュメントのソースを、17行目で右のフレームに表示するHTMLドキュメントを生成するためのCGIリクエストを、それぞれ埋め込むようになっている。

【0161】同図中、17行目のCGIパラメータの最後に付加されている、`rid=<<EMBED NAME = “__RID__” >>`という一連のトークンが、CG<sup>50</sup>

28

Iリクエストを識別するためのダミーパラメータを付加するためのスクリプトであり、前記図15および図16に示した一連の処理にしたがってトークンが解析され、HTMLドキュメントが生成される。

【0162】図18は、本発明のネットワークデバイス管理ソフトウェアにおけるHTMLドキュメント生成手段が生成した、HTMLドキュメントの一例である。

【0163】同図に示したのは、前記図17のテンプレートの解析結果であり、前記図17の17行目に記述されているダミーパラメータを埋め込むための一連のトークン、

```
rid=<<EMBED NAME = “__RID__” >>
```

の解析結果は、図10の10行目にあるように、`rid=884`となっている。

【0164】以上のように、本実施形態のネットワーク管理装置および方法によれば、デバイスからMIB情報を取得し、Webブラウザからのリクエストに応じて、HTMLドキュメントを生成し、前記HTMLドキュメント中にCGIリクエストを埋め込む場合に、当該CGIリクエストのパラメータの1つとして、HTMLドキュメントを生成することに異なる値をダミーパラメータとして付加することにより、Webブラウザのキャッシュ機能が有効になっていた場合であっても、表示が更新される可能性があるWebページにアクセスする場合に、Webブラウザのキャッシュに保存されたHTMLドキュメントではなく、Webサーバと通信して取得した新しいHTMLドキュメントが表示されるようになる。そのため、表示の更新が行われないWebページと、表示の更新が行われる可能性があるページとで、選択的にブラウザのキャッシュに保存されたHTMLドキュメントを表示するかどうかを制御することが可能になるという効果がある。

【0165】これにより、例えば、設定系のWebページで値を変更した後、表示系のWebページで前記変更した値を閲覧するような場合に、表示系のWebページを表示する際にブラウザのキャッシュに保存されている古いHTMLドキュメントが表示されてしまうことにより、前記設定系のWebページで変更したはずの最新の値と異なる値が表示されてしまうという不都合を回避することが可能になる。

【0166】なお、本発明は、複数の機器（例えばホストコンピュータ、インタフェース機器、リーダなど）から構成されるシステムあるいは統合装置に適用しても、ひとつの機器からなる装置に適用してもよい。

【0167】また、前述した実施形態の機能を実現するための、図5及び図6の手順のプログラムコードを記録した記憶媒体を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ（またはCPUや



MPU)が記憶媒体に格納されたプログラムコードを読み出し実行することによっても、本発明の目的が達成される。

【0168】この場合、記憶媒体から読み出されたプログラムコード自体が本発明の新規な機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。

【0169】プログラムコードを供給するための記憶媒体としては、例えば、フロッピーディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CDD-R、磁気テープ、不揮発性のメモ리카ード、ROMなどを用いることができる。

【0170】また、コンピュータが読み出したプログラムコードを実行することによって、前述した実施形態の機能が実現される他、そのプログラムコードの指示に基づき、コンピュータ上で稼動しているOSなどが実際の処理の一部または全部を行い、その処理によっても前述した実施形態の機能が実現され得る。

【0171】さらに、記憶媒体から読み出されたプログラムコードが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書き込まれた後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるCPUなどが実際の処理の一部または全部を行い、その処理によっても前述した実施形態の機能が実現され得る。

【0172】なお、本発明は、前述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体から、そのプログラムをパソコン通信など通信ラインを介して要求者にそのプログラムを配信する場合にも適用できる。

【0173】

【発明の効果】以上説明したように、本発明によれば、デバイスのMIB情報の保守を行う場合、ユーザによる変更が可能なMIB情報については、デバイスから読み出したMIB情報もクライアントが変更不可能な形式でクライアントに送ることで、変更されたMIB情報に限り、デバイスに設定することができる。これにより、デバイスとの間のトラフィックの増加や、デバイスの負荷の増加を防止するとともに、設定ページ中に存在する設定形式オブジェクトの構成を管理する必要をなくすることで、HTMLドキュメントあるいはそのテンプレート等の変更が容易に行えるネットワーク管理装置及び方法を提供できる。

【0174】また、WebブラウザからMIB情報を変更させるためのHTMLドキュメント生成する際、その中にCGIリクエストを埋め込む場合に、当該CGIリクエストのパラメータの1つとして、当該CGIリクエストを固有に識別するためのパラメータを埋め込むことにより、Webブラウザのキャッシュ機能が有効な場

合であっても、キャッシュに保存されたHTMLドキュメントではなく、Webサーバから取得した新しいHTMLドキュメントを表示させることができる。

【図面の簡単な説明】

【図1】プリンタをネットワークに接続するためのネットワークボードを、開放型アーキテクチャをもつプリンタへつなげた場合を示す図である。

【図2】一般的なWWWシステムの動作およびWWWシステムに基づいたSNMP管理プログラムの動作の概要を説明する図である。

【図3】ネットワークデバイス管理ソフトウェアが稼動可能なPCの構成を示すブロック図である。

【図4】ネットワークデバイス管理ソフトウェアのモジュール構成図である。

【図5】第1実施形態のネットワークデバイス管理ソフトウェアにおいて、CGIリクエストに応じて、HTMLドキュメントを生成する処理の一例を示すフローチャートである。

【図6】第1実施形態のネットワークデバイス管理ソフトウェアにおいて、CGIパラメータで通知されたMIB情報をデバイスに設定する処理の一例を示すフローチャートである。

【図7】ネットワークに接続された管理対象デバイスを一覧表示する、デバイスリストの表示例を示す図である。

【図8】デバイスのより詳細な情報を表示するためのデバイス詳細の表示例を示す図である。

【図9】デバイスの情報を変更するための設定ページの表示例を示す図である。

【図10】第1実施形態のネットワークデバイス管理ソフトウェアが生成したHTMLドキュメントの一例を示す図である。

【図11】本発明におけるネットワークデバイス管理ソフトウェアが記憶された記憶媒体の一例であるCD-ROMのメモリマップを示すイメージ図である。

【図12】第2実施形態のネットワークデバイス管理ソフトウェアにおいて、CGIリクエストに応じて、HTMLドキュメントを生成する処理の一例を示すフローチャートである。

【図13】第2実施形態のネットワークデバイス管理ソフトウェアにおいて、HTML文書に埋め込まれる、ダミーパラメータを生成するためのスクリプトの一例を示す図である。

【図14】第2実施形態のネットワークデバイス管理ソフトウェアにおいて、HTML文書に埋め込まれる、ダミーパラメータを含むURLを有したページへ分岐するためのスクリプトの一例を示す図である。

【図15】第3実施形態のネットワークデバイス管理ソフトウェアにおいて、CGIリクエストに応じて、HTMLドキュメントを生成する処理の一例を示すフローチャートである。

31

ャートである。

【図16】第3実施形態のネットワークデバイス管理ソフトウェアにおいて、HTML文書に埋め込むダミーパラメータを生成する処理の一例を示すフローチャートである。

【図17】第3実施形態のネットワークデバイス管理ソフトウェアにより使用されるHTMLドキュメントのテンプレート一例を示す図である。

【図18】図17のテンプレートから作成されるHTMLドキュメントの一例を示す図である。

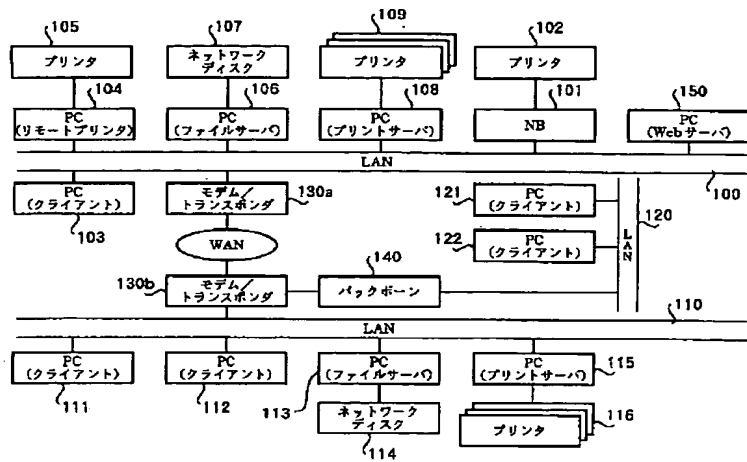
【符号の説明】

100 ローカルエリアネットワーク (LAN)  
 101 ネットワークボード (NB)  
 102 開放型アーキテクチャをもつプリンタ  
 103 LAN100に接続されているパーソナルコンピュータ (PC)  
 104 LAN100に接続されているパーソナルコンピュータ (PC)  
 105 PC104に接続されているプリンタ  
 106 LAN100に接続されているファイルサーバ<sup>20</sup>  
 107 LAN100に含まれるネットワークディスク  
 108 LAN100に含まれるプリントサーバ  
 109 プリントサーバ108に接続されたプリンタ  
 110 ローカルエリアネットワーク (LAN)  
 111 LAN110に接続されているパーソナルコンピュータ (PC)  
 112 LAN110に接続されているパーソナルコンピュータ (PC)  
 113 LAN110に接続されているファイルサーバ  
 114 LAN110に含まれるネットワークディスク<sup>30</sup>  
 115 LAN110に含まれるプリントサーバ  
 116 プリントサーバ110に接続されたプリンタ  
 117 プリントサーバ110に接続されたプリンタ  
 120 ローカルエリアネットワーク (LAN)

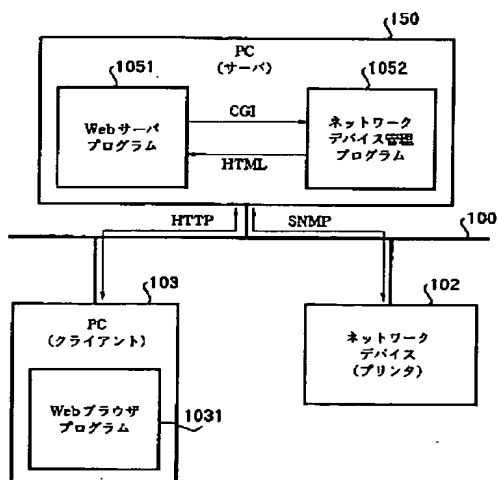
32

121 LAN120に接続されているパーソナルコンピュータ (PC)  
 122 LAN120に接続されているパーソナルコンピュータ (PC)  
 130 モデム/トランスポンダ  
 140 バックボーン  
 150 LAN100に接続されているWebサーバ  
 200 共有メモリ  
 301 CPU  
 302 CPU301の動作プログラムを格納するためのROM  
 303 ワークエリアとして用いるためのRAM  
 304 システムバス  
 305 キーボードコントローラ  
 306 CRTコントローラ  
 307 ディスクコントローラ  
 308 ネットワークインタフェースカード (NIC)  
 309 キーボード  
 310 CRT  
 311 ハードディスク (HD)  
 312 フロッピーディスク (FD)  
 402 CGIインタフェース  
 403 全体制御モジュール  
 404 パラメータモジュール  
 405 システムモジュール  
 406 システム設定ファイル  
 407 デバイスリストモジュール  
 408 デバイス探索モジュール  
 409 プロトコルモジュール  
 410 デバイス詳細モジュール  
 411 デバイス固有モジュール  
 412 テンプレートモジュール  
 413 テンプレートファイル  
 414 キャッシュファイル

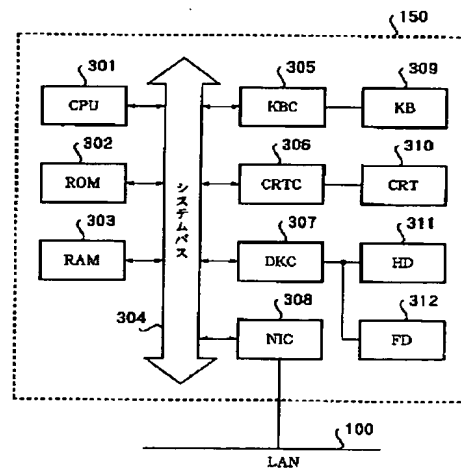
【図1】



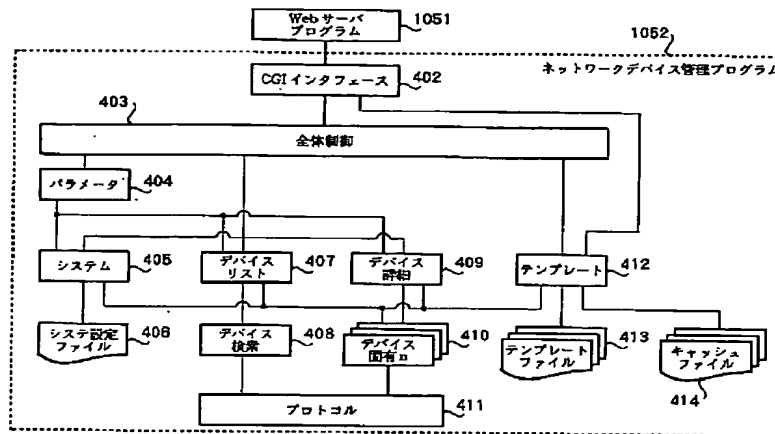
【図2】



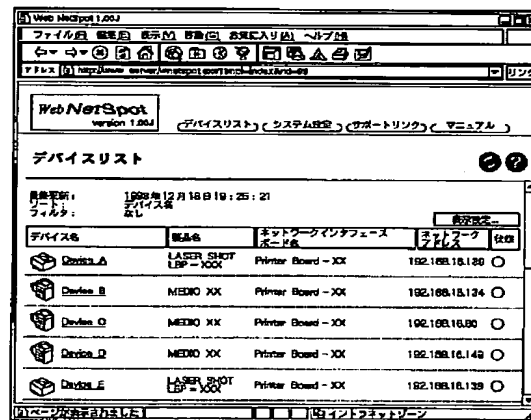
【図3】



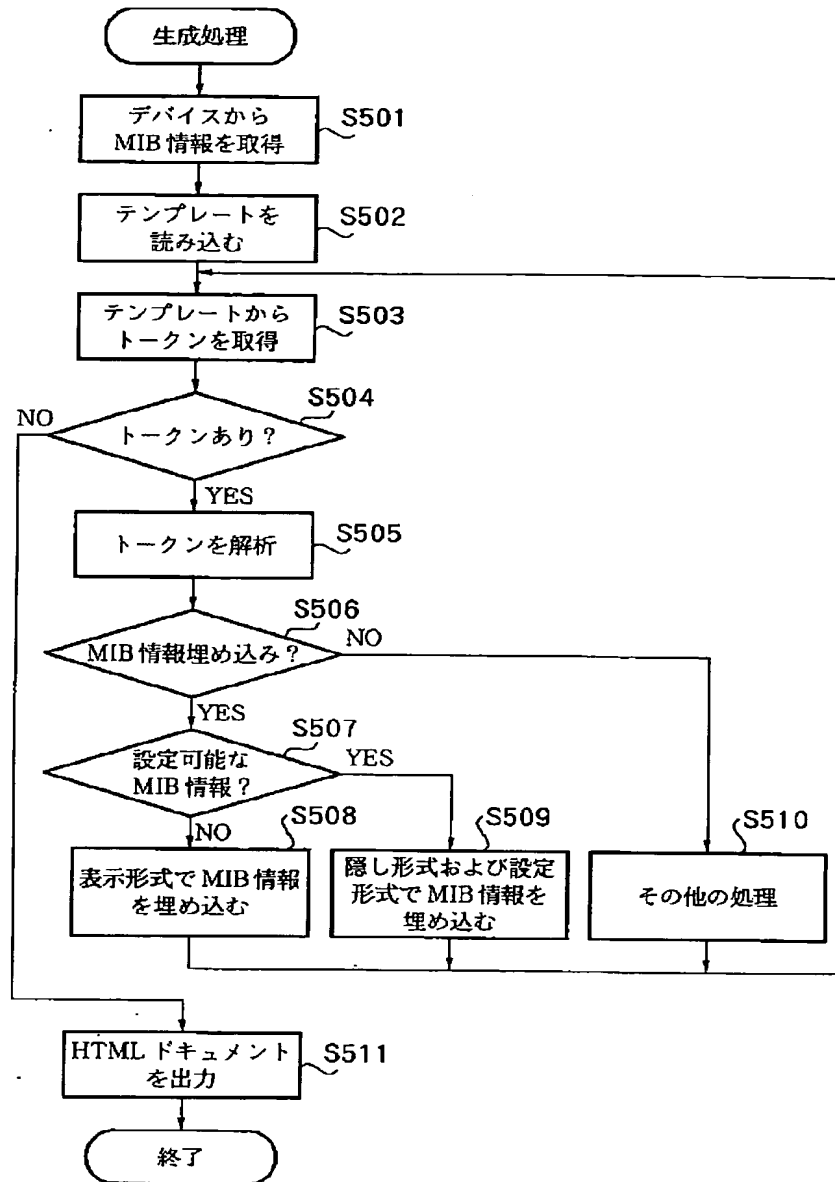
【図4】



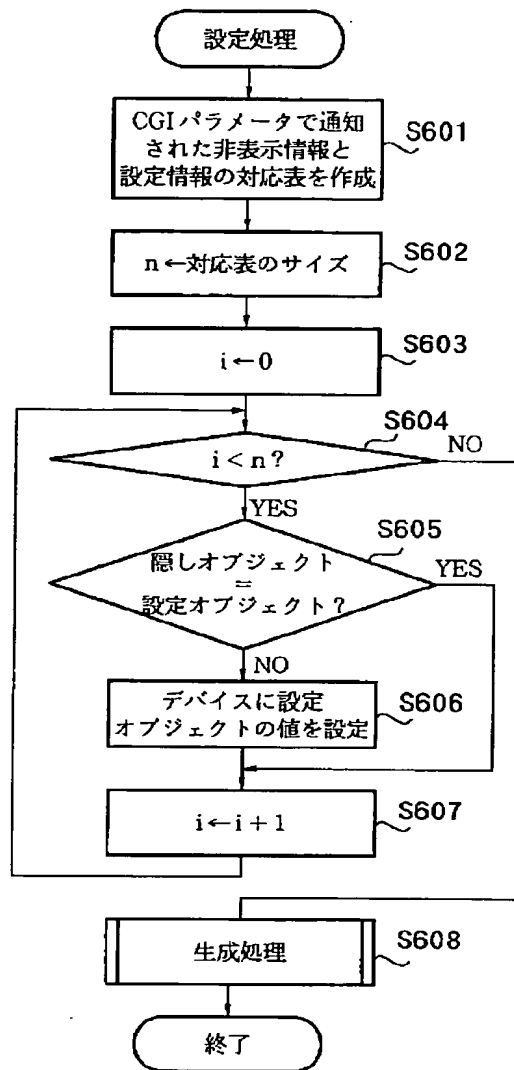
【図7】



【図5】



【図6】



[illegible][illegible]

【図10】

```

00: <FORM NAME="PrinterSetup" ACTION="POST">
01:
02: <INPUT TYPE="hidden" NAME="InputSelected" VALUE="1">
03: <INPUT TYPE="hidden" NAME="InputAutoFeed" VALUE="1">
04: <INPUT TYPE="hidden" NAME="InputAutoFeed" VALUE="3">
05: <INPUT TYPE="hidden" NAME="PreferenceTray" VALUE="1">
06: <INPUT TYPE="hidden" NAME="TrayPaperSize" VALUE="1">
07: <INPUT TYPE="hidden" NAME="EnvelopeSize" VALUE="1">
08:
09: <TABLE BORDER="0" CELSPACING="4" CELLSPACING="2" WIDTH="100%">
10: <TR><TD WIDTH="100%"><P>
11: <TABLE BORDER="0" CELSPACING="0" CELLSPACING="2" WIDTH="100%">
12: <TR HEIGHT="15">
13: <TD BGCOLOR="#666666" WIDTH="99%" HEIGHT="15">
14: <B><FONT COLOR="#FFFFFF"> 印刷機 </FONT></B></TD>
15: </TR>
16: </TABLE></TD>
17: </TR>
18: <TR>
19: <TD><TABLE BORDER="0" CELSPACING="4" CELLSPACING="2" WIDTH="100%">
20: <TR><TD WIDTH="16">&nbsp;</TD>
21: <TD NOWRAP>印刷機: </TD>
22: <TD COLSPAN="2" WIDTH="89%">
23: <SELECT NAME="InputSelected">
24: <OPTION VALUE="1" SELECTED>印刷機トレイ
25: <OPTION VALUE="2">上段カセット
26: <OPTION VALUE="3">下段カセット
27: <OPTION VALUE="4">ペーパーチャージ
28: <OPTION VALUE="5">別冊フイード
29: </SELECT></TD>
30: </TR>
31: <TR VALIGN="top">
32: <TD WIDTH="15">&nbsp;</TD>
33: <TD WIDTH="15">&nbsp;</TD>
34: <TD NOWRAP>自動装紙: </TD>
35: <TD WIDTH="59%">
36: <SELECT NAME="InputAutoFeed" SIZE="4" MULTIPLE>
37: <OPTION VALUE="1" SELECTED>装紙トレイ
38: <OPTION VALUE="2">上段カセット
39: <OPTION VALUE="3" SELECTED>下段カセット
40: <OPTION VALUE="4">ペーパーチャージ
41: </SELECT></TD>
42: </TR>
43: <TR VALIGN="top">
44: <TD WIDTH="15">&nbsp;</TD>
45: <TD WIDTH="15">&nbsp;</TD>
46: <TD COLSPAN="2" NOWRAP>INPUT TYPE="checkbox" VALUE="1" NAME="PreferenceTray">トレイ優先</TD>
47: </TR>
48: <TR VALIGN="top">
49: <TD WIDTH="16">&nbsp;</TD>
50: <TD WIDTH="16">&nbsp;</TD>
51: <TD NOWRAP>トレイ印刷サイズ: </TD>
52: <TD>
53: <SELECT NAME="Input" SIZE="1">
54: <OPTION VALUE="1">A3
55: <OPTION VALUE="2">A4
56: <OPTION VALUE="3" SELECTED>レター
57: </SELECT></TD>
58: </TR>
59: <TR VALIGN="top">
60: <TD WIDTH="16">&nbsp;</TD>
61: <TD WIDTH="16">&nbsp;</TD>
62: <TD NOWRAP>封筒サイズ: </TD>
63: <TD>
64: <SELECT NAME="EnvelopeSize" SIZE="0">
65: <OPTION VALUE="1" SELECTED>ユースペーパー
66: <OPTION VALUE="2">封筒2号
67: <OPTION VALUE="3">封筒4号
68: </SELECT></TD>
69: </TR>
70: </TABLE></TD>
71: </TABLE>
72:
73: </FORM>

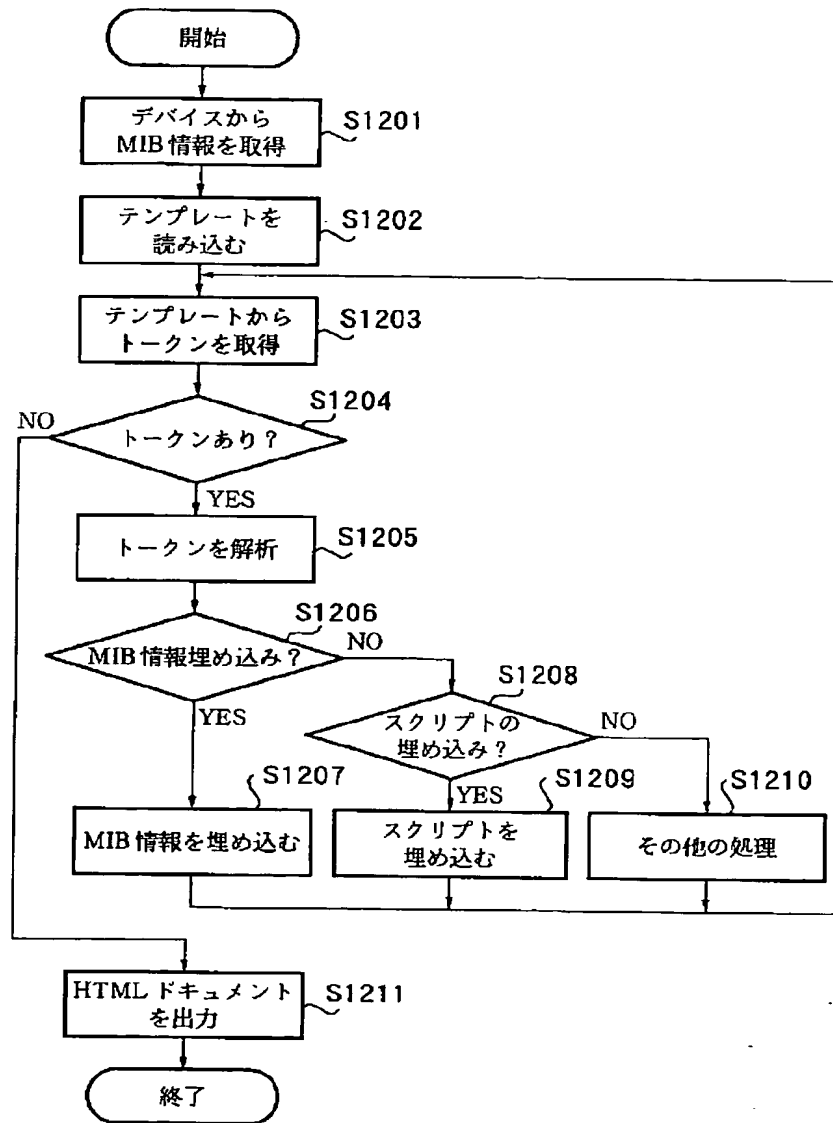
```



【図 1 1】

ディレクトリ情報	9999
...	
インストールプログラム	9998
...	
ネットワークデバイス 制御プログラム	9997
...	

【図12】



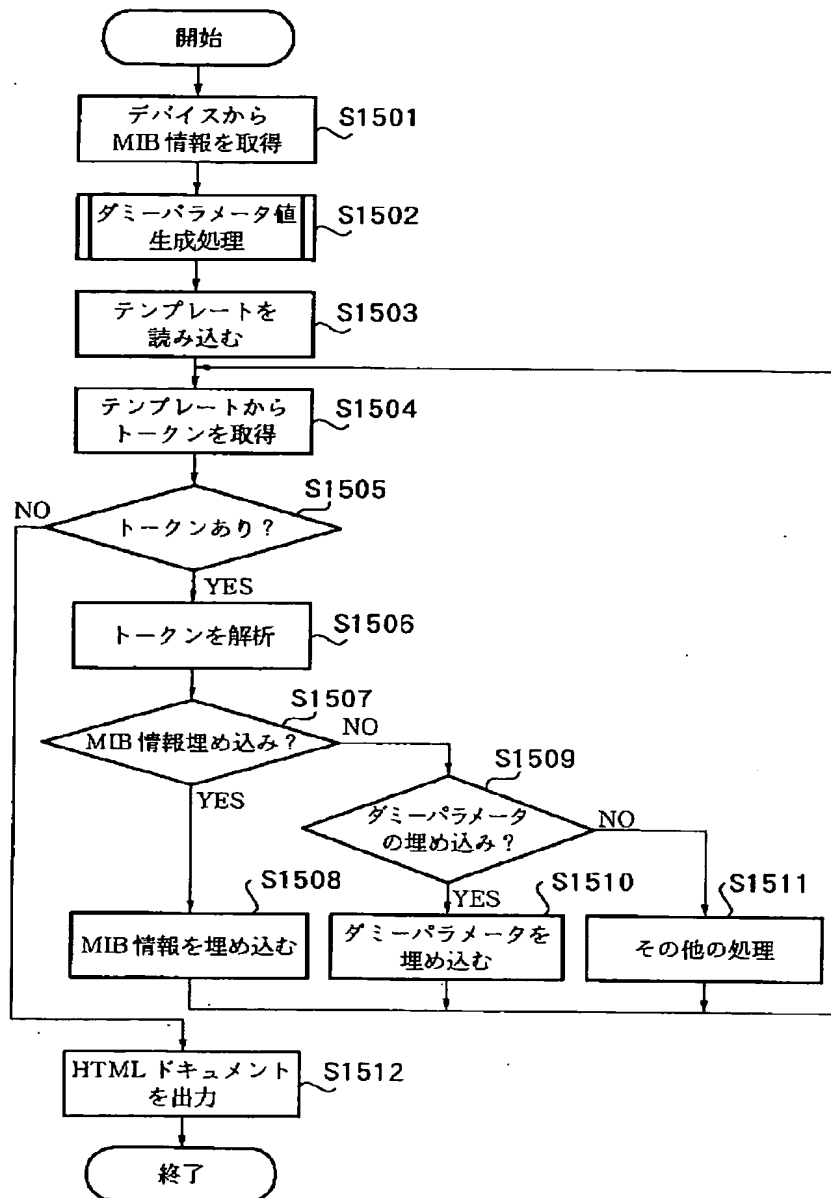
【図13】

```
00: <SCRIPT LANGUAGE="JavaScript" >
01:
02: function getObj (addr, tmpl)
03: {
04:     var now = new Date ();
05:
06:     var url = 'wnewspt.exe'
07:         + '&addr=' + addr
08:         + '&tmpl=' + tmpl
09:         + '&rid=' + now.getTime ();
10:
11:     return url;
12: }
13:
14: </SCRIPT>
```

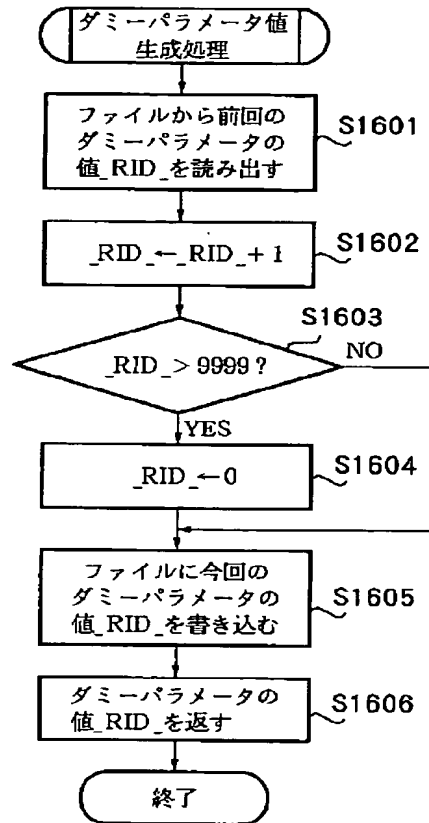
【図14】

```
00: function gotoDeviceDetail (addr)
01: {
02:   top.location = gotoGj (addr, "default");
03: }
```

【図15】



【図16】



【図17】

```

00: <<TEMPLATE>>
01:
02: <<HEAD>>
03: <</HEAD>>
04:
05: <<BODY>>
06: Content-type: text/html
07:
08: <HTML>
09:
10: <HEAD>
11: <META HTTP-EQUIV='content-type' CONTENT='text/html; charset = x-ujis' >
12: <TITLE>Web NetSpot 1.00</TITLE>
13: </HEAD>
14:
15: <FRAMESET COLS='140,*' FRAMESORDER='NO' BORDER='0' FRAMESPACING='0' >
16: <FRAME SRC='html/srs/detailmenu.html' MARGINWIDTH='8' MARGINHEIGHT='0' NAME='Menu' NORESIZE SCROLLING='NO' >
17: <FRAME SRC='wristspot_ers?addr=<<EMBED NAME='NETADDR' >>&tmpl=body frame&rid=<<EMBED NAME='RID' >>' NAME='Body' >
18: </FRAMESET>
19:
20: <NOFRAMES>
21: <BODY BGCOLOR='#FFFFFF' >
22: </BODY>
23: </NOFRAMES>
24:
25: </HTML>
26: <</BODY>>
27:
28: <</TEMPLATE>>

```

【図18】

```
00: < HTML >
01:
02: < HEAD >
03:   < META HTTP-EQUIV = ' content-type ' CONTENT = ' text/html ; charset = x-øjis >
04:   < TITLE > Web NeSpot 1.00 < /TITLE >
05: < /HEAD >
06:
07: < FRAMESET COLS = ' 140,* ' FRAMEORDER = ' NO ' BORDER = ' 0 ' FRAMESPACING = ' 0 ' >
08:   < FRAME SRC = ' html/svs/defaultmenu.html ' MARGINWIDTH = ' 0 ' MARGINHEIGHT = ' 0 ' NAME = ' Menu ' NORESIZE SCROLLING = ' NO ' >
09:   < FRAME SRC = ' wnetspot_exe?cmd = devget&addr = 192.168.16.139&tmpl = body_frame&rid = 884 ' NAME = ' Body ' >
10: < /FRAMESET >
11:
12:
13: < NOFRAMES >
14:   < BODY BGCOLOR = ' #FFFFFF ' >
15: < /BODY >
16: < /NOFRAMES >
17:
18: < /HTML >
```